



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/639,943	08/13/2003	Richard H. Boivic	YOR920030260US1 (16780)	6976
23389	7590	11/21/2006	EXAMINER LANIER, BENJAMIN E	
SCULLY SCOTT MURPHY & PRESSER, PC 400 GARDEN CITY PLAZA SUITE 300 GARDEN CITY, NY 11530			ART UNIT 2132	PAPER NUMBER

DATE MAILED: 11/21/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/639,943	Applicant(s) BOIVIE ET AL.	
	Examiner Benjamin E. Lanier	Art Unit 2132	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☐ Responsive to communication(s) filed on ____.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-32 is/are pending in the application.
- 4a) Of the above claim(s) ____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) ____ is/are allowed.
- 6) ☒ Claim(s) 1-4 and 12-32 is/are rejected.
- 7) ☒ Claim(s) 5-11 is/are objected to.
- 8) ☐ Claim(s) ____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 14 October 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. ____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|--|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. ____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date ____ | 6) <input type="checkbox"/> Other: ____ |

DETAILED ACTION

Claim Objections

1. A series of singular dependent claims is permissible in which a dependent claim refers to a preceding claim which, in turn, refers to another preceding claim.
2. A claim which depends from a dependent claim should not be separated by any claim which does not also depend from said dependent claim. It should be kept in mind that a dependent claim may refer to any preceding independent claim. In general, applicant's sequence will not be changed. See MPEP § 608.01(n).
3. Claim 5 depends from claim 3 and is separated by claim 4, which is not dependent upon claim 3. Claims 5-11 are objected to.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.
5. Claims 1, 14, 23, 25, 31 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.
6. Claim 1 recites the limitation "said integrity values" in line 10. There is insufficient antecedent basis for this limitation in the claim. Only a single integrity value had been defined previously.
7. Claim 14 recites the limitation "said stored integrity values" in line 11. There is insufficient antecedent basis for this limitation in the claim. Only a single integrity value had been defined previously.

Art Unit: 2132

8. Claim 23 recites the limitation "a written data block's version number and checksum" in lines 2-3. There is insufficient antecedent basis for this limitation in the claim. A checksum was never defined as part of a data blocks data structure in the tree. For the purposes of examination, previously defined 'integrity value' will used instead.

9. Claim 25 recites the limitation "said stored integrity values" in line 13. There is insufficient antecedent basis for this limitation in the claim. Only a single integrity value had been defined previously.

10. Claim 31 recites the limitation "a written data block's version number and checksum" in line 3. There is insufficient antecedent basis for this limitation in the claim. A checksum was never defined as part of a data blocks data structure in the tree. For the purposes of examination, previously defined 'integrity value' will used instead.

Claim Rejections - 35 USC § 102

11. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

12. Claims 1, 2, 12, 14, 15, 18-26, 28-32 are rejected under 35 U.S.C. 102(e) as being anticipated by Foster, U.S. Publication No. 2003/0200448.

The applied reference has a common inventor and assignee with the instant application. Based upon the earlier effective U.S. filing date of the reference, it constitutes prior art under 35 U.S.C. 102(e). This rejection under 35 U.S.C. 102(e) might be overcome either by a showing

under 37 CFR 1.132 that any invention disclosed but not claimed in the reference was derived from the inventor of this application and is thus not the invention “by another,” or by an appropriate showing under 37 CFR 1.131.

Referring to claim 1, Foster discloses storing encrypted data in a memory ([0056] & Figures 4, 7-9), which meets the limitation of a storage device for storing encrypted data. An encryption means (Figure 6A, element 249) performs encryption on the data prior to being stored on the memory ([0072]), which meets the limitation of means at a client device for encrypting data prior to writing data blocks at said storage device. The encryption capabilities of the data access control function prevent direct observation and modification of data, the integrity check function adds the ability to further verify that the encrypted value is the same value that was written to memory originally ([0050]), which meets the limitation of said encryption means employing encryption capable of protecting individual data blocks against modification, relocation and replay for each data block written to said storage device. A first integrity check value that is a mathematically condensed version of the data to be secured and authenticated is generated, encrypted, and stored with the data in the memory ([0054]-[0056]), which meets the limitation of means for generating an integrity value corresponding to one or more data blocks, and integrity value comprising information for preventing modification of data for each data block written to said storage device, means for storing said integrity values of written data blocks. Element 249 of Figure 6A & 6B provides means of encryption and decryption for the data and integrity check values ([0070]-[0075]), which meets the limitation of means at said client device for decrypting said encrypted data accessed from said storage device. The encrypted data is retrieved from memory, decrypted, and an integrity value is calculated for the

decrypted data block ([0070]). The integrity value retrieved from external memory and the integrity value calculated from the retrieved and decrypted data are compared ([0071]). If the integrity values match, then the data is considered authenticated and is returned to the requestor ([0071]). If the integrity values do not match, then the integrity check function may return an error indicator used as part of a tamper detection function ([0071]), which meets the limitation of means for performing an integrity check at said client device utilizing stored integrity values corresponding to stored data blocks being accessed, wherein the integrity check protects the integrity of contents stored in said storage device.

Referring to claim 2, Foster discloses that the encryption is performed with a key ([0065]), which meets the limitation of said encryption means generates encrypted cipher text data blocks that are a function of plaintext data included in said data block and a first encryption key.

Referring to claim 12, Foster discloses that the memory is non-volatile ([0037]), which meets the limitation of said storage device comprises a non-volatile storage device.

Referring to claim 14, Foster discloses storing encrypted data in a memory ([0056] & Figures 4, 7-9), which meets the limitation of a storage device for storing encrypted data. An encryption means (Figure 6A, element 249) performs encryption on the data prior to being stored on the memory ([0072]), which meets the limitation of encrypting data to be written from a client device to storage device for storing encrypted data. The encryption capabilities of the data access control function prevent direct observation and modification of data, the integrity check function adds the ability to further verify that the encrypted value is the same value that was written to memory originally ([0050]), which meets the limitation of said encryption utilizing an encryption

scheme capable of protecting individual data blocks against modification, relocation and replay for each data block written to said storage device. A first integrity check value that is a mathematically condensed version of the data to be secured and authenticated is generated, encrypted, and stored with the data in the memory ([0054]-[0056]), which meets the limitation of generating an integrity value corresponding to one or more data blocks, and integrity value comprising information for preventing modification of data for each data block written to said storage device, storing said integrity values of written data blocks. Element 249 of Figure 6A & 6B provides means of encryption and decryption for the data and integrity check values ([0070]-[0075]), which meets the limitation of decrypting the encrypted data accessed from said storage device. The encrypted data is retrieved from memory, decrypted, and an integrity value is calculated for the decrypted data block ([0070]). The integrity value retrieved from external memory and the integrity value calculated from the retrieved and decrypted data are compared ([0071]). If the integrity values match, then the data is considered authenticated and is returned to the requestor ([0071]). If the integrity values do not match, then the integrity check function may return an error indicator used as part of a tamper detection function ([0071]), which meets the limitation of performing an integrity check at said client device utilizing stored integrity values corresponding to stored data blocks being accessed, wherein the integrity check protects the integrity of contents stored in said storage device.

Referring to claim 15, Foster discloses that the encryption is performed with a key ([0065]), which meets the limitation of said encrypting data step includes generating cipher text data blocks that are a function of plaintext data included in said data block and a first encryption key.

Referring to claim 18, Foster discloses generating integrity tree structure for storing integrity values corresponding to each disk block written to said storage device (Figure 12 & [0086]-[0087]).

Referring to claim 19, Foster discloses that the integrity value can be used in a hierarchical manner to be able to authenticate a large address range with a single integrity value (Figure 11 & [0084]). Integrity values can be generated for a series of data blocks ([0084]). For example, if an integrity value is one-eighth the size of the data, and if integrity values are generated for 8 data blocks, it will yield a collection of 8 integrity values that are equal in size to one data block ([0084]). This collection of integrity values can then be considered a block of data to be authenticated, and another layer of integrity value is generated that authenticates the previous layer of 8 integrity values ([0084]), which meets the limitation of said integrity tree structure comprises hierarchical data structure, said hierarchical data structure including two or more layers of integrity data structures, each successive layer of integrity data structures including meta-data protecting integrity of data at an immediate prior layer.

Referring to claim 20, Foster discloses that large address regions can be authenticated with a single integrity value through the use of multiple layers of integrity values (Figure 12 & [0086]). The integrity value is one-fourth the size of the data block, a root integrity value can be used to authenticate a next four integrity values, which are labeled level 1 (Figure 12 & [0086]). Each of these integrity values in turn authenticates another four integrity values, labeled in level 2 (Figure 12 & [0086]). This process continues until the level n-1 integrity values authenticate the actual data blocks stored in level n (Figure 12 & [0086]), which meets the limitation of writing encrypted data blocks at a first layer of said hierarchical data structure, and writing a

Art Unit: 2132

succeeding layer of meta-data blocks, each meta-data block including data structures representing a plurality of disk blocks written at said first layer, each meta-data block data structure comprising an integrity value and a version number pair for each of said plurality of disk blocks.

Referring to claim 21, Foster discloses that large address regions can be authenticated with a single integrity value through the use of multiple layers of integrity values (Figure 12 & [0086]). The integrity value is one-fourth the size of the data block, a root integrity value can be used to authenticate a next four integrity values, which are labeled level 1 (Figure 12 & [0086]). Each of these integrity values in turn authenticates another four integrity values, labeled in level 2 (Figure 12 & [0086]). This process continues until the level n-1 integrity values authenticate the actual data blocks stored in level n (Figure 12 & [0086]), which meets the limitation of writing a succeeding layer of higher-level meta-data blocks for protecting a layer of meta-data blocks below, each higher level meta-data block comprising data structure representing a plurality of meta-data blocks, each higher level meta-data block data structure comprising an integrity value and version number pair for each of said plurality of meta-data blocks.

Referring to claim 22, Foster discloses a root integrity value can be used to authenticate a next four integrity values, which are labeled level 1 (Figure 12 & [0086]). Each of these integrity values in turn authenticates another four integrity values, labeled in level 2 (Figure 12 & [0086]). This process continues until the level n-1 integrity values authenticate the actual data blocks stored in level n (Figure 12 & [0086]), which meets the limitation of generating a root data structure at a top layer of said hierarchical data structure for protecting integrity of all content written to said storage device.

Referring to claim 23, Foster discloses that when a write request for data is received the integrity values all the way up to the root are recalculated ([0089]), which meets the limitation of writing a data block to said storage device, said writing including updating a written data block's version number and checksum in the associated meta-data blocks, and, said checksum and version number value updating being performed at each successive meta-data layer corresponding to said written data block, including updating performed at said root data structure.

Referring to claim 24, Foster discloses that an integrity value is calculated for requested data and compared with all integrity values in the tree until the requested data is reached ([0088]), which meets the limitation of reading a data block from said storage device, performing an integrity check including comparing integrity of data blocks to be read on a path from said root data structure via successive meta-data block layers until a desired data block is read from said first layer of said hierarchical data structure.

Referring to claim 25, Foster discloses storing encrypted data in a memory ([0056] & Figures 4, 7-9), which meets the limitation of a storage device for storing encrypted data. An encryption means (Figure 6A, element 249) performs encryption on the data prior to being stored on the memory ([0072]), which meets the limitation of encrypting data to be written from a client device to storage device for storing encrypted data. The encryption capabilities of the data access control function prevent direct observation and modification of data, the integrity check function adds the ability to further verify that the encrypted value is the same value that was written to memory originally ([0050]), which meets the limitation of said encryption utilizing an encryption scheme capable of protecting individual data blocks against modification, relocation and replay

Art Unit: 2132

for each data block written to said storage device. A first integrity check value that is a mathematically condensed version of the data to be secured and authenticated is generated, encrypted, and stored with the data in the memory ([0054]-[0056]), which meets the limitation of generating an integrity value corresponding to one or more data blocks, and integrity value comprising information for preventing modification of data for each data block written to said storage device, storing said integrity values of written data blocks. Element 249 of Figure 6A & 6B provides means of encryption and decryption for the data and integrity check values ([0070]-[0075]), which meets the limitation of decrypting the encrypted data accessed from said storage device. The encrypted data is retrieved from memory, decrypted, and an integrity value is calculated for the decrypted data block ([0070]). The integrity value retrieved from external memory and the integrity value calculated from the retrieved and decrypted data are compared ([0071]). If the integrity values match, then the data is considered authenticated and is returned to the requestor ([0071]). If the integrity values do not match, then the integrity check function may return an error indicator used as part of a tamper detection function ([0071]), which meets the limitation of performing an integrity check at said client device utilizing stored integrity values corresponding to stored data blocks being accessed, wherein the integrity check protects the integrity of contents stored in said storage device.

Referring to claim 26, Foster discloses that the encryption is performed with a key ([0065]), which meets the limitation of said encrypting data step includes generating cipher text data blocks that are a function of plaintext data included in said data block and a first encryption key.

Referring to claim 28, Foster discloses that the integrity value can be used in a hierarchical manner to be able to authenticate a large address range with a single integrity value (Figure 11 & [0084]). Integrity values can be generated for a series of data blocks ([0084]). For example, if an integrity value is one-eighth the size of the data, and if integrity values are generated for 8 data blocks, it will yield a collection of 8 integrity values that are equal in size to one data block ([0084]). This collection of integrity values can then be considered a block of data to be authenticated, and another layer of integrity value is generated that authenticates the previous layer of 8 integrity values ([0084]), which meets the limitation of said integrity tree structure comprises hierarchical data structure, said hierarchical data structure including two or more layers of integrity data structures, each successive layer of integrity data structures including meta-data protecting integrity of data at an immediate prior layer.

Referring to claim 29, Foster discloses that large address regions can be authenticated with a single integrity value through the use of multiple layers of integrity values (Figure 12 & [0086]). The integrity value is one-fourth the size of the data block, a root integrity value can be used to authenticate a next four integrity values, which are labeled level 1 (Figure 12 & [0086]). Each of these integrity values in turn authenticates another four integrity values, labeled in level 2 (Figure 12 & [0086]). This process continues until the level n-1 integrity values authenticate the actual data blocks stored in level n (Figure 12 & [0086]), which meets the limitation of writing encrypted data blocks at a first layer of said hierarchical data structure, and writing a succeeding layer of meta-data blocks, each meta-data block including data structures representing a plurality of disk blocks written at said first layer, each meta-data block data

structure comprising an integrity value and a version number pair for each of said plurality of disk blocks.

Referring to claim 30, Foster discloses that large address regions can be authenticated with a single integrity value through the use of multiple layers of integrity values (Figure 12 & [0086]). The integrity value is one-fourth the size of the data block, a root integrity value can be used to authenticate a next four integrity values, which are labeled level 1 (Figure 12 & [0086]). Each of these integrity values in turn authenticates another four integrity values, labeled in level 2 (Figure 12 & [0086]). This process continues until the level n-1 integrity values authenticate the actual data blocks stored in level n (Figure 12 & [0086]), which meets the limitation of writing a succeeding layer of higher-level meta-data blocks for protecting a layer of meta-data blocks below, each higher level meta-data block comprising data structure representing a plurality of meta-data blocks, each higher level meta-data block data structure comprising an integrity value and version number pair for each of said plurality of meta-data blocks. A root integrity value can be used to authenticate a next four integrity values, which are labeled level 1 (Figure 12 & [0086]). Each of these integrity values in turn authenticates another four integrity values, labeled in level 2 (Figure 12 & [0086]). This process continues until the level n-1 integrity values authenticate the actual data blocks stored in level n (Figure 12 & [0086]), which meets the limitation of generating a root data structure at a top layer of said hierarchical data structure for protecting integrity of all content written to said storage device.

Referring to claim 31, Foster discloses that when a write request for data is received the integrity values all the way up to the root are recalculated ([0089]), which meets the limitation of writing a data block to said storage device, said writing including updating a written data block's

Art Unit: 2132

version number and checksum in the associated meta-data blocks, and, said checksum and version number value updating being performed at each successive meta-data layer corresponding to said written data block, including updating performed at said root data structure.

Referring to claim 32, Foster discloses that an integrity value is calculated for requested data and compared with all integrity values in the tree until the requested data is reached ([0088]), which meets the limitation of reading a data block from said storage device, performing an integrity check including comparing integrity of data blocks to be read on a path from said root data structure via successive meta-data block layers until a desired data block is read from said first layer of said hierarchical data structure.

13. Claims 1-3, 12, 14-16, 25-27 are rejected under 35 U.S.C. 102(e) as being anticipated by Foster, U.S. Patent No. 6,715,085.

The applied reference has a common assignee and inventor with the instant application. Based upon the earlier effective U.S. filing date of the reference, it constitutes prior art under 35 U.S.C. 102(e). This rejection under 35 U.S.C. 102(e) might be overcome either by a showing under 37 CFR 1.132 that any invention disclosed but not claimed in the reference was derived from the inventor of this application and is thus not the invention "by another," or by an appropriate showing under 37 CFR 1.131.

Referring to claim 1, Foster discloses storing encrypted data in a memory (Col. 7, lines 13-14), which meets the limitation of a storage device for storing encrypted data. An encryption means (Figure 7A, element 249) performs encryption on the data prior to being stored on the memory (Col. 7, lines 11-12), which meets the limitation of means at a client device for

encrypting data prior to writing data blocks at said storage device. The encryption capabilities of the data access control function prevent direct observation and modification of data, the integrity check function adds the ability to further verify that the encrypted value is the same value that was written to memory originally (Col. 6, lines 44-60), which meets the limitation of said encryption means employing encryption capable of protecting individual data blocks against modification, relocation and replay for each data block written to said storage device. A first integrity check value that is a mathematically condensed version of the data to be secured and authenticated is generated, encrypted, and stored with the data in the memory (Col. 7, lines 7-14), which meets the limitation of means for generating an integrity value corresponding to one or more data blocks, and integrity value comprising information for preventing modification of data for each data block written to said storage device, means for storing said integrity values of written data blocks. Element 249 of Figure 7A provides means of encryption and decryption for the data and integrity check values (Col. 7, lines 16-17), which meets the limitation of means at said client device for decrypting said encrypted data accessed from said storage device. The encrypted data is retrieved from memory, decrypted, and an integrity value is calculated for the decrypted data block (Col. 7, lines 18-20). The integrity value retrieved from external memory and the integrity value calculated from the retrieved and decrypted data are compared (Col. 7, lines 21-24). If the integrity values match, then the data is considered authenticated and is returned to the requestor (Col. 7, lines 21-24). If the integrity values do not match, then the action is taken (Col. 7, lines 21-24), which meets the limitation of means for performing an integrity check at said client device utilizing stored integrity values corresponding to stored data

Art Unit: 2132

blocks being accessed, wherein the integrity check protects the integrity of contents stored in said storage device.

Referring to claim 2, Foster discloses that the encryption is performed with a key (Col. 9, lines 3-4), which meets the limitation of said encryption means generates encrypted cipher text data blocks that are a function of plaintext data included in said data block and a first encryption key.

Referring to claim 3, Foster discloses that the data is encrypted using a key, the memory address, and a version number for whitening (Col. 9, lines 30-32 & Col. 10, lines 30-38), which meets the limitation of said encryption means implements a whitening value which is a function of a second encryption key, an address location for said storage block, and a version number indicating a block write increment, said encryption means further generating cipher text data blocks that are additionally a function of said whitening value.

Referring to claim 12, Foster discloses that the external memory is non-volatile (Col. 4, lines 51), which meets the limitation of said storage device comprises a non-volatile or volatile storage device.

Referring to claim 14, Foster discloses storing encrypted data in a memory (Col. 7, lines 13-14), which meets the limitation of a storage device for storing encrypted data. An encryption means (Figure 7A, element 249) performs encryption on the data prior to being stored on the memory (Col. 7, lines 11-12), which meets the limitation of encrypting data to be written from a client device to storage device for storing encrypted data. The encryption capabilities of the data access control function prevent direct observation and modification of data, the integrity check function adds the ability to further verify that the encrypted value is the same value that was

Art Unit: 2132

written to memory originally (Col. 6, lines 44-60), which meets the limitation of said encryption utilizing an encryption scheme capable of protecting individual data blocks against modification, relocation and replay for each data block written to said storage device. A first integrity check value that is a mathematically condensed version of the data to be secured and authenticated is generated, encrypted, and stored with the data in the memory (Col. 7, lines 7-14), which meets the limitation of generating an integrity value corresponding to one or more data blocks, and integrity value comprising information for preventing modification of data for each data block written to said storage device, storing said integrity values of written data blocks. Element 249 of Figure 7A provides means of encryption and decryption for the data and integrity check values (Col. 7, lines 16-17), which meets the limitation of decrypting the encrypted data accessed from said storage device. The encrypted data is retrieved from memory, decrypted, and an integrity value is calculated for the decrypted data block (Col. 7, lines 18-20). The integrity value retrieved from external memory and the integrity value calculated from the retrieved and decrypted data are compared (Col. 7, lines 21-24). If the integrity values match, then the data is considered authenticated and is returned to the requestor (Col. 7, lines 21-24). If the integrity values do not match, then the integrity check function may return an error indicator used as part of a tamper detection function (Col. 7, lines 21-24), which meets the limitation of performing an integrity check at said client device utilizing stored integrity values corresponding to stored data blocks being accessed, wherein the integrity check protects the integrity of contents stored in said storage device.

Referring to claim 15, Foster discloses that the encryption is performed with a key (Col. 9, lines 3-4), which meets the limitation of said encrypting data step further includes generating

encrypted cipher text data blocks that are a function of plaintext data included in said data block and a first encryption key.

Referring to claim 16, Foster discloses that the data is encrypted using a key, the memory address, and a version number for whitening (Col. 9, lines 30-32 & Col. 10, lines 30-38), which meets the limitation of said encrypting data step further includes generating a whitening value as a function of a second encryption key, an address location for said storage block, and a version number indicating a block write increment, and generation of cipher text data blocks that are additionally a function of said whitening value.

Referring to claim 25, Foster discloses storing encrypted data in a memory (Col. 7, lines 13-14), which meets the limitation of a storage device for storing encrypted data. An encryption means (Figure 7A, element 249) performs encryption on the data prior to being stored on the memory (Col. 7, lines 11-12), which meets the limitation of encrypting data to be written from a client device to storage device for storing encrypted data. The encryption capabilities of the data access control function prevent direct observation and modification of data, the integrity check function adds the ability to further verify that the encrypted value is the same value that was written to memory originally (Col. 6, lines 44-60), which meets the limitation of said encryption utilizing an encryption scheme capable of protecting individual data blocks against modification, relocation and replay for each data block written to said storage device. A first integrity check value that is a mathematically condensed version of the data to be secured and authenticated is generated, encrypted, and stored with the data in the memory (Col. 7, lines 7-14), which meets the limitation of generating an integrity value corresponding to one or more data blocks, and integrity value comprising information for preventing modification of data for each data block

written to said storage device, storing said integrity values of written data blocks. Element 249 of Figure 7A provides means of encryption and decryption for the data and integrity check values (Col. 7, lines 16-17), which meets the limitation of decrypting the encrypted data accessed from said storage device. The encrypted data is retrieved from memory, decrypted, and an integrity value is calculated for the decrypted data block (Col. 7, lines 18-20). The integrity value retrieved from external memory and the integrity value calculated from the retrieved and decrypted data are compared (Col. 7, lines 21-24). If the integrity values match, then the data is considered authenticated and is returned to the requestor (Col. 7, lines 21-24). If the integrity values do not match, then the integrity check function may return an error indicator used as part of a tamper detection function (Col. 7, lines 21-24), which meets the limitation of performing an integrity check at said client device utilizing stored integrity values corresponding to stored data blocks being accessed, wherein the integrity check protects the integrity of contents stored in said storage device.

Referring to claim 26, Foster discloses that the encryption is performed with a key (Col. 9, lines 3-4), which meets the limitation of said encrypting data step further includes generating encrypted cipher text data blocks that are a function of plaintext data included in said data block and a first encryption key.

Referring to claim 27, Foster discloses that the data is encrypted using a key, the memory address, and a version number for whitening (Col. 9, lines 30-32 & Col. 10, lines 30-38), which meets the limitation of said encrypting data step further includes generating a whitening value as a function of a second encryption key, an address location for said storage block, and a version

Art Unit: 2132

number indicating a block write increment, and generation of cipher text data blocks that are additionally a function of said whitening value.

Claim Rejections - 35 USC § 103

14. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

15. The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

16. Claims 4, 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Foster, U.S. Publication No. 2003/0200448, in view of Aiello, U.S. Patent No. 5,608,801. Referring to claims 4, 17, Foster discloses that encryption is used to protect the data ([0009]), but does not disclose what particular encryption scheme is used. It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize the DES algorithm to encrypt the data of Foster because DES provides a reasonable fast and commercially available encryption algorithm as taught in Aiello (Col. 3, lines 55-57).

17. Claims 4, 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Foster, U.S. Patent No. 6,715,085, in view of Aiello, U.S. Patent No. 5,608,801. Referring to claims 4, 17,

Art Unit: 2132

Foster discloses that encryption is used to protect the data (Col. 7, lines 7-14), but does not disclose what particular encryption scheme is used. It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize the DES algorithm to encrypt the data of Foster because DES provides a reasonable fast and commercially available encryption algorithm as taught in Aiello (Col. 3, lines 55-57).

18. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Foster, U.S. Publication No. 2003/0200448, in view of Sit, U.S. Patent No. 6,898,707. Referring to claim 13, Foster discloses verifying encrypted data that is stored in a memory ([0056], [0070]-[0075] & Figures 4, 7-9). Foster discloses that the memory may be external ([0037]), but does not specify that the memory is remotely connected over a network. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the memory remotely connected over a network so that users in an organizational network can have access to a single data verifier as taught in Sit (Figure 8 & Col. 4, line 56 – Col. 5, line 3).

19. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Foster, U.S. Patent No. 6,715,085, in view of Sit, U.S. Patent No. 6,898,707. Referring to claim 13, Foster discloses verifying encrypted data that is stored in a memory (Col. 7, lines 7-24). Foster discloses that the memory may be external (Col. 4, lines 49-51), but does not specify that the memory is remotely connected over a network. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the memory remotely connected over a network so that users in an organizational network can have access to a single data verifier as taught in Sit (Figure 8 & Col. 4, line 56 – Col. 5, line 3).

Conclusion

Art Unit: 2132

20. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Taki, U.S. Publication No. 2003/0159037

Brundrett, U.S. Patent No. 6,249,866

Herbert, U.S. Patent No. 6,708,274

Carpentier, U.S. Patent No. 6,976,165

Asano, U.S. Publication No. 2002/0169971

21. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Benjamin E. Lanier whose telephone number is 571-272-3805. The examiner can normally be reached on M-Th 7:30am-5:00pm, F 7:30am-4pm.

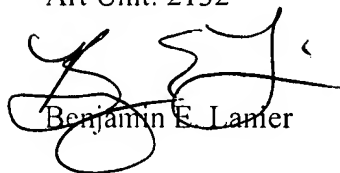
If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gilberto Barron can be reached on 571-272-3799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

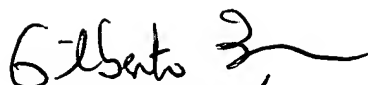
Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Application/Control Number: 10/639,943

Page 22

Art Unit: 2132


Benjamin E. Lanier


GILBERTO BARRON JR
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100